

# Fernkurs-Dokumentation



## Zentrum für Informatik ZFI AG

**Fernkurs: Oracle Certified Professional, Java  
SE 7 Programmierer II, 1Z0-804 (FOCP-1017) -IT**

### **Ausbildung nach Mass**

<http://www.zfi.ch/FOCP-1017>

Weitere Infos finden Sie unter [www.zfi.ch](http://www.zfi.ch) oder via Adresse:

**Zentrum für Informatik ZFI AG  
Zentralsekretariat  
Rütistrasse 28  
CH-8952 Zürich-Schlieren  
Telefon: 044 732 40 00  
Telefax: 041 530 31 68**

**Zürich, Basel, Bern, Zürich, Schweiz**

<b>Titel</b>	<b>Fernkurs: Oracle Certified Professional, Java SE 7 Programmierer II, 1Z0-804</b>
<b>Untertitel</b>	<b>Der Vorbereitungslehrgang auf die offizielle Oracle Certified Professional, Java SE 7 Programmierer II 1Z0-804 Prüfung als Fernkurs.</b>
<b>Einleitung</b>	<p>Die Oracle Certified Professional (OCP), Java SE 7 Programmierer II Zertifizierung richtet sich an Software-Entwickler, welche ihre Kenntnisse der Java Programmiersprache attestiert haben möchten. Das Bestehen der Prüfung Java Programmierer II (1Z0-804) bestätigt, dass der Programmierer den Syntax und die Struktur der Java-Programmiersprache versteht und mit Java Applikationen entwickeln kann, die auf Server- oder Desktop-Systemen mit Java SE 7 laufen.</p> <p>Mit der Variante Fernkurs erarbeiten Sie sich die theoretischen Grundlagen selbständig und festigen Ihr Wissen anhand von vielen Übungen. Die wichtigsten Fakten werden in einem Factsheet zusammengefasst mitgeliefert und sind ein wichtiger Teil der Prüfungsvorbereitung.</p> <p>Eine begleitende grössere Programmierübung rundet den Fernkurs ab. Bei Fragen können Sie den ZFI Kursleiter per EMail kontaktieren und Ihre Fragen werden umgehend und kompetent beantwortet.</p> <p>Als Entwicklungsumgebung setzen wir die Eclipse Java IDE ein. Sie können aber auch die Oracle Netbeans IDE oder IntelliJ verwenden.</p> <p>Die erfolgreiche Absolvierung der OCA Prüfung 1Z0-803 (siehe FOCA) wird für diese Prüfung 1Z0-804, Oracle Certified Professional, Java Programmierer II, vorausgesetzt.</p> <p>Nach der Anmeldung für diesen Fernkurs erhalten Sie von uns das Begleitbuch per Post zugestellt. Zugleich werden Sie von uns per EMail kontaktiert und Sie erhalten von uns den Link auf die elektronischen Unterlagen (Dokument, Übungen und Beispiele) für den Download. In einer ersten Phase arbeiten Sie die Unterlagen inkl. den Übungen durch. Die Übungen enthalten Musterlösungen. Eine Übungsreihe gilt als Repetition, die Lösung senden Sie per EMail an uns zur Kontrolle. Sie können uns jederzeit über den vereinbarten Mail-Account kontaktieren, wir helfen Ihnen gerne weiter. In einer 2. Phase erfolgt das Training auf die Prüfung selbst indem Sie viele echte Prüfungsfragen beantworten. Zu diesen Fragen gibt es wiederum Lösungen. Planen Sie für diese Vorbereitung mehrere Wochen ein, dann sind Sie für die Prüfung bereit und optimal vorbereitet.</p> <p>Die aktuellen Java SE 7/8 Programmierer Zertifizierungen von Oracle:</p> <ul style="list-style-type: none"><li>Java SE 7 Programmierer I (1Z0-803)</li><li>Java SE 7 Programmierer II (1Z0-804)</li><li>Upgrade to Java SE 7 Programmierer (1Z0-805)</li><li>Java SE 8 Programmierer I (1Z0-808)</li><li>Java SE 8 Programmierer II (1Z0-809)</li></ul>
<b>Ihr Nutzen</b>	<b>Dieser ZFI-Lehrgang richtet sich an Personen, welche die Sprache Java systematisch lernen möchten, um anschliessend Java-Applikationen in</b>

<b>Voraussetzungen</b>	ihrem beruflichen Umfeld zu entwickeln und zu warten. Lernziel ist selbstverständlich das Bestehen der von Oracle durchgeführten Prüfung zum Oracle Certified Professional Java Programmer II (1Z0_804).
<b>Teilnehmerkreis</b>	Dieser ZFI-Lehrgang richtet sich an Programmierer, welche bereits erste Schritte in Java gemacht haben oder bereits in einer anderen Programmiersprache entwickelt haben und die Grundlagen der objektorientierten Software-Entwicklung bereits kennen.
<b>Unterlagen</b>	<ul style="list-style-type: none"><li>- Begleitbuch</li><li>- Tutorials</li><li>- Intranet Site</li></ul>
<b>Folgekurse</b>	
<b>Inhalt</b>	<b>Oracle Certified Professional, Java SE 7 Programmer II</b> <ul style="list-style-type: none"><li>- Java Class Design</li><li>- Use access modifiers: private, protected, and public</li><li>- Override methods</li><li>- Overload constructors and other methods appropriately</li><li>- Use the instanceof operator and casting</li><li>- Use virtual method invocation</li><li>- Override methods from the Object class to improve the functionality of your class</li><li>- Use package and import statements</li><li>- Advanced Class Design</li><li>- Identify when and how to apply abstract classes</li><li>- Construct abstract Java classes and subclasses</li><li>- Use the static and final keywords</li><li>- Create top-level and nested classes</li><li>- Use enumerated types</li></ul>

- **Object-Oriented Design Principles**
- **Write code that declares, implements and/or extends interfaces**
- **Choose between interface inheritance and class inheritance**
- **Develop code that implements "is-a" and/or "has-a" relationships.**
- **Apply object composition principles**
- **Design a class using the Singleton design pattern**
- **Write code to implement the DAO pattern**
- **Design and create objects using a factory, and use factories from the API**
  
- **Generics and Collections**
- **Create a generic class**
- **Use the diamond syntax to create a collection**
- **Analyze the interoperability of collections that use raw type and generic types**
- **Use wrapper classes and autoboxing**
- **Create and use a List, a Set and a Deque**
- **Create and use a Map**
- **Use java.util.Comparator and java.lang.Comparable**
- **Sort and search arrays and lists**
  
- **String Processing**

- Search, parse and build strings
- Search, parse, and replace strings by using regular expressions
- Use string formatting
  
- Exceptions and Assertions
  
- Use throw and throws statements
- Use the try statement with multi-catch, and finally clauses
- Autoclose resources with a try-with-resources statement
- Create custom exceptions
- Test invariants by using assertions
  
- Java I/O Fundamentals
  
- Read and write data from the console
- Use streams to read and write files
  
- Java File I/O (NIO.2)
  
- Use the Path class to operate on file and directory paths
- Use the Files class to check, delete, copy, or move a file or directory
- Read and change file and directory attributes
- Recursively access a directory tree

- Find a file by using the PathMatcher class
- Watch a directory for changes by using WatchService
  
- Building Database Applications with JDBC
- Define the layout of the JDBC API
- Connect to a database by using a JDBC driver
- Update and query a database
- Customize the transaction behavior of JDBC and commit transactions
- Use the JDBC 4.1 RowSetProvider, RowSetFactory, and RowSet interfaces
  
- Threads
- Create and use the Thread class and the Runnable interface
- Manage and control thread lifecycle
- Synchronize thread access to shared data
- Identify potential threading problems
  
- Concurrency
- Use java.util.concurrent collections
- Apply atomic variables and locks
- Use Executors and ThreadPools
- Use the parallel Fork/Join Framework

- Localization
- Read and set the locale by using the Locale object
- Build a resource bundle for each local
- Load a resource bundle in an application
- Format text for localization by using NumberFormat and DateFormat

**Beitrag**

Der Teilnehmerbeitrag versteht sich rein netto. Das ZFI ist (gemäss MwSt-Gesetz) nicht Mehrwertsteuerpflichtig und erhebt somit keine MwSt. Bei länger als einen Monat dauernden Lehrgängen ist die Zahlung des Teilnehmerbeitrages in mehreren Raten möglich (pro rata temporis).